1. Background
2. Methodology
3. Matched Filter Algorithm
4. Experimental Results
5. Future Work
6. Poster

Background
Introduction to the License Plate Recognition project.

Introduction

License plate recognition (LPR) presents an interesting image processing challenge with a relevant, real world application. While LPR technology exists for highway ticketing and police monitoring systems, expanding the functionality to more small scale applications can increase the versatility and affordability of the technology. Utilizing Matlab's image processing capabilities, our team aims to harness the power of these available functions to develop a matched filter algorithm, and test the degree of accuracy in which it can identify the characters on a license plate.

Motivation

The inspiration for the project came from a parking technology project that some of our group members have been working on called ParkiT. Through conversations with parking garage operators throughout Houston, the team identified License Plate Recognition as one of the most valuable products. The technology is versatile and can be applied for a variety functions such as security, automatic entry and payment, and better management of the facility. Through this project our team will develop a foundation for an LPR algorithm, which can be improved upon and developed into a fully functioning system that can be tested in a garage trial next semester. More information about ParkiT can be found at www.parkitfast.com.

Methodology
Overview of matched filter technique used to identify characters on a license plate.

Introduction

While the human eye can quickly identify characters on a plate and compare for similarity, training a program to use computer vision to do the same task presents many new difficulties. First, the program must be able to identify the license plate out of the captured data image, crop the image to obtain the important information, normalize the size and positioning of the plate, and finally apply a matched filter algorithm to match each character to the correct characters from a template. Each step of the process presents a unique challenge. In this section, we will cover the subtleties of each step and the solutions we chose to address them.

Project Overview

1. Data Collection and Template Dictionary

The first challenge in developing a system that can accurately identify a license plate is to be able to recognize a license in a given image. This step is certainly the most challenging part, and requires the use of an advanced signal processing algorithm known as the Scale Invariant Feature Transform (SIFT). Even though we will not be discussing the algorithm in detail as it is not the focus of this module, it is necessary to understand how it can be used to provide added functionality to the identification system that we are presenting. For a given image, the SIFT algorithm relies on interesting points of a given object to be able to recognize it. These points provide a "feature description" of the object, and can be extracted from a training image. For the algorithm to work properly, the features extracted must be recognizable under changes in illumination, scale, and image noise. Once the object of interest is identified on an image, it can then be manipulated, and rescaled to an optimal size and position for the processing required during identification.

The method presented in this module assumes that the license has already been identified and placed in a position suitable for the matched filter. In

order for the tests to faithfully represent how the system will operate in a real world application, it is necessary that the input images be all taken from the same angle. In addition, the system presented here does not require that the images used be of high quality. Any camera can achieve adequate results, however, higher resolution can improve efficiency of the program. For our testing and data collection, we used an iPhone camera to accomplish this.

Template Dictionary Character ‘B’



Image of template dictionary character ‘B’ used by matched filter algorithm.

Test License Plate



Image of Texas License Plate used for testing.

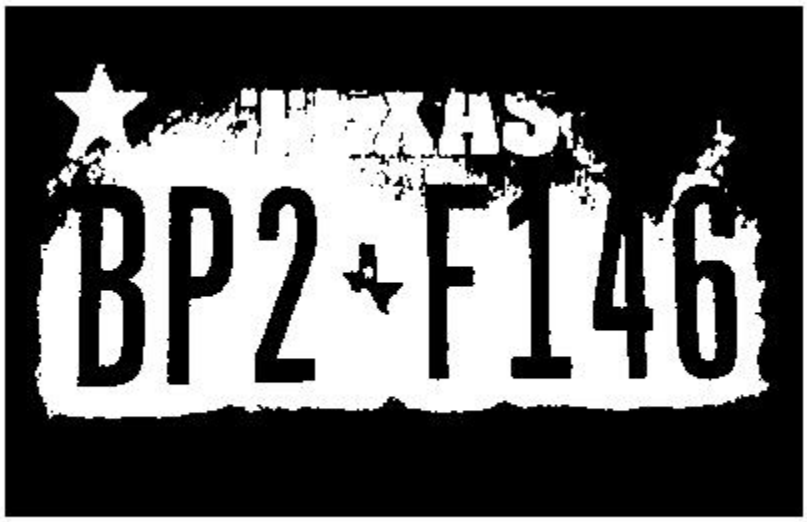
2. Image Normalization

The next step is crucial to the image processing of the license plate. The matched filter is a sensitive algorithm which relies heavily on the resemblance between the characters on the license plate and the template images. Consequently, it is necessary that the template characters and the images of the license plate whose characters we want to recognize are standardized. Therefore, whenever an image of a license plate is input into the function, it is automatically rescaled to 256 by 400 pixels. The reason these dimensions were chosen is that it allows us to efficiently run the correlation between many templates, and the image of the license plate. Furthermore, the individual template dictionary characters are all re-scaled to 110 by 40 pixels. This choice is critical to the success of the algorithm as it preserves the ratio of the width to height of the characters.

After the size of the images is adjusted, all the images (both license plate and templates) are changed to black and white images. This ensure that all the pixels have a binary value of either zero or one, and this step is crucial to ensure the accuracy of the matched filter. Finally, the last step in the

normalization process is to subtract a value of 0.5 from every pixel in the image. This step is performed to avoid the dilemma of obtaining a correlation result of zero in regions where there is a match. Even though an area might have pixel values of zeroes, it could be matching another area with similar pixel values, however, because the pixel values are zero, the result of the correlation comes to be zero as well. By subtracting a value of 0.5, we ensure that two areas that do match will both have either negative values or positive values; therefore, the correlation will yield a high positive score.

Black and White License Plate



License plate image converted to black and white.

Rescaled License Plate



Black and White License Plate rescaled by 0.5. Black pixels rescaled from 0 to 0.5. White pixels rescaled from 1 to -0.5.

3. Matched Filter Template Comparison

Now that the experimental components are set up, the matched filter can be applied. We first section the license plate to obtain sections with only one character on them. Once the sectioning is complete, we then perform the correlation between each section, and all the template images, and for each correlation, we save the scores of the correlation in an array. Once the algorithm is done running on a specific section, we compare the scores obtained, and pick the character associated with the highest score as the match at the current location.

Matched Filter Algorithm
Summary of the matched filter algorithm procedure.

The Template Dictionary

The matched filter template dictionary is a vital component to the success of the program. This dictionary contains a sample of each character, A-Z and 1-9. Each image is obtained from an actual license plate image, converted to black and white, and resized to 256x400 pixels to reduce processing time. Our library of individual character samples was the result of cropping the characters out of several different license plate templates. We finally stored the template dictionary as a cell in MATLAB so we could run the program immediately when we input a license plate image.

Matched Filter Algorithm

In order to identify characters on the license plate, we must run our test image through a series of steps (Figure 1).

1. Image Normalization
First, to match the template dictionary samples, every license plate is resized to the standard 256 by 400 pixels. Then, it is processed into a black and white image, with the pixel value 0 corresponding to black and pixel value 1 corresponding to white. This was done to set up for a 2D correlation between the two images.

2. Image Partition
We then partition the image into 7 different sections, one for each individual character of the license plate. This was done to make the computation more efficient, as we will only compute the correlation between our template dictionary, which contains our library of all of the license plate characters, and the sections of the license plate that contain each individual character.

3. Rescale Pixel Values
Our next step is to shift the pixel values from [0,1] to [-0.5,0.5]. When we compute the correlation between the partitions of the license plate and the

template characters, this will ensure we obtain only positive values if the images are the same, and negative values in that region if they are different.

4. Two-Dimensional Correlation

To check for matches between our two images, we perform a correlation between the test license plate and the each entry in the template dictionary. To set this up, our program rescales the pixel values by calculating 1/2 - each pixel value. Now the pixels to range from 1/2 (black) to -1/2 (white). Next we flip the template character upside down and right to left and run the Matlab 'conv2' function. This is equivalent to a 2D correlation between the two images. Two negative numbers (black and black) will return a positive result and two positive numbers (white and white) will return a positive result as well. Through this method, we can obtain all the matches between the two images.

Equation for 2D Convolution

$$(g * f)(x, y) = \sum_{(a,b) \in A} g(a, b) f(x - a, y - b)$$

Equation for 2D Discrete Convolution.

Matched Filter Algorithm

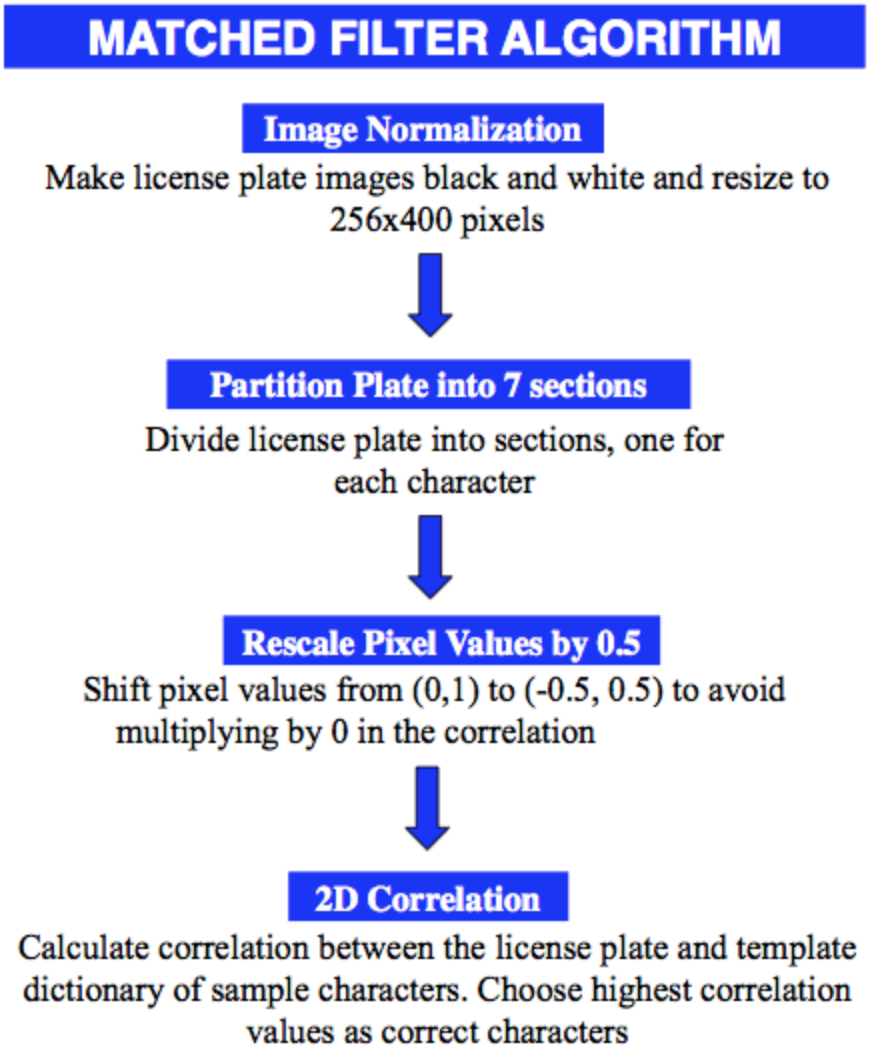


Diagram of Matched Filter Algorithm for License Plate Recognition.

Program Results

On each license plate, we section the entire image to obtain each character (110x40). After running a convolution between all the test image sections and the template dictionary, we find the highest correlation values to determine the right character. Our program will return the identified

characters in a string, in order of their location on the license plate. This string should match the characters on the actual license plate.

Experimental Results
Results of experimentation with matched filter technique.

Trial 1 Results

The first run through of our program showed promising results for the numbers, but less than stellar results for some of the letters. After several rounds of experimentation, we identified several ways to improve our algorithm. One major change was standard cropping for the template characters and the test license plate. While our original template dictionary images were cropped from the original images downloaded from the camera phone, our test images were obtained from a 256 by 400 pixel crop, thus there were slight differences in the images which lead to results that were not passing the matched filter test. Because matched filter is a very sensitive test, it responds to even the slightest changes that may not be detectable by a quick glance from a human eye.

Another adjustment that improved our results was flipping the image right to left and upside down before applying the 'conv2' function. This set up our image so that applying the function was equivalent to calculating a correlation, yielding significantly more sensitive and accurate results.

Trial 2 Results

After changing our template dictionary samples so our dictionary and test images were cropped from a 256 by 400 pixel image and making adjustments to the image before applying ‘conv2’, we saw much better results. Our program accurately identifies characters on this type of Texas plate, returning the correct characters in order. Thus far it has produced accurate results on the five plates we have tested on, with an average processing time of 0.78252 seconds.

For one of our test images, license plate 'BP2F146' (Figure 1) we tested the template dictionary character '4' with the entire license plate. The results showed that the program found the location and match of '4' on the right place in the license plate, as shown in a heat map in Figure 2. This visual also shows the low correlation, represented by cooler colors, of the other

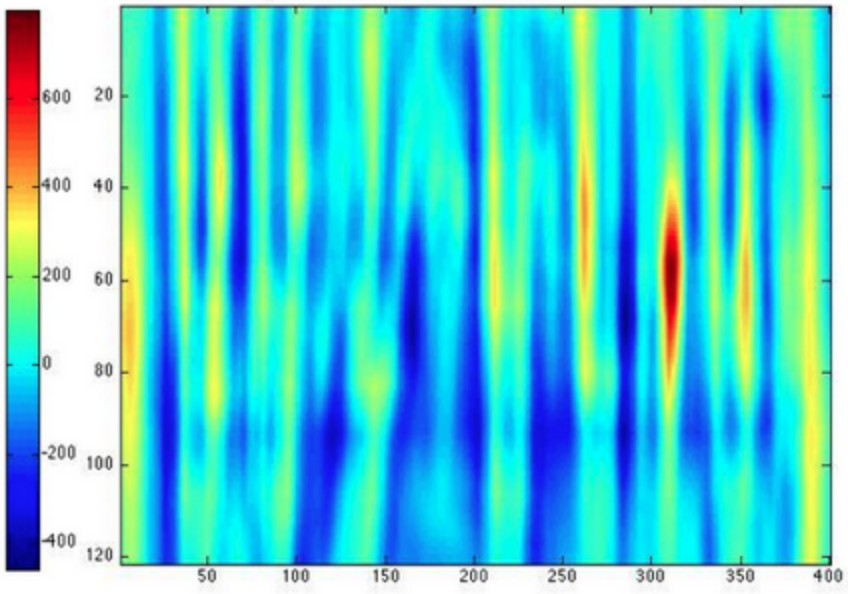
pixels of the license plate with the template '4' character. Figure 3 shows the results of correlating position six (sixth character) on the license plate with the entire template dictionary. In this case, the sixth character is '4' and the stem plot shows that indeed the template '4' was chosen as the highest match and most probable choice. However, our experiments did reveal that other characters do not produce as distinct correlation results and may not offer an obvious match choice. When correlation values are close, our program will always choose the highest one which could possibly lead to an incorrect detection.

Test License Plate



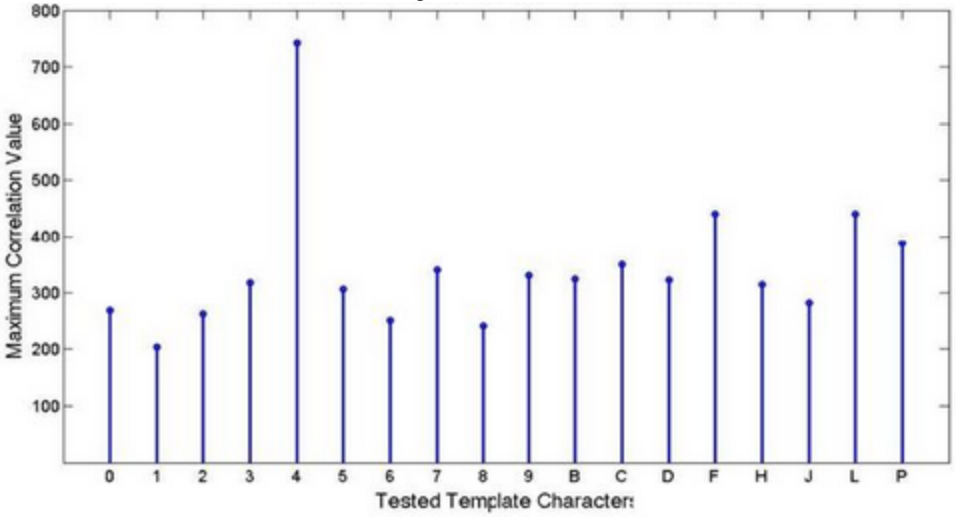
Image of Texas License Plate used for testing.

Template Character '4' Identified and Located on License Plate



Correlation of license plate with template character ‘4’ produces this heat map with correct location of '4' at position six.

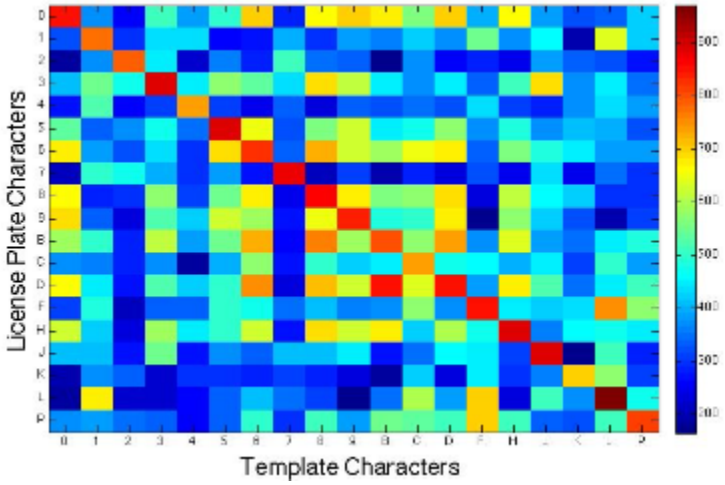
'4' Identified as Most Likely Match at Position Six



Correlation results of position six on the test license plate with all the template characters, shows ‘4’ as the match.

To further illustrate the degree of accuracy of our program, we have constructed a Confusion Matrix of the correlation results (Figure 4) between license plate characters and the template dictionary characters. After running our program on several test license plates, we obtain the average correlation value between each character and the template dictionary to create this data visualization that shows the highest matches in warm colors. Through this diagram, it can be seen that the close matches lie along the diagonal of the matrix, which is ideal as the diagonal represents the matching characters between the template dictionary and license plate. However, on this figure it is also easy to visualize the "trouble characters", such as the '8', 'B', and the 'D', which are slightly off the diagonal and may show similar high correlation values even when they are not the correct match.

Visualization of High Correlation Values Between Matches



Confusion Matrix illustrating the highest matches with warm colors.

Future Work
Result of experiments and future work that can be done.

LPR Challenges

While our program works efficiently for a subset of license plates, there are still various challenges that we must overcome to obtain a more robust program. These include: differences between different states' license plates, trouble characters such as "B" and "8", and environmental adaptation for a real world application. In this module, we will suggest potential avenues to pursue in order to address these challenges.

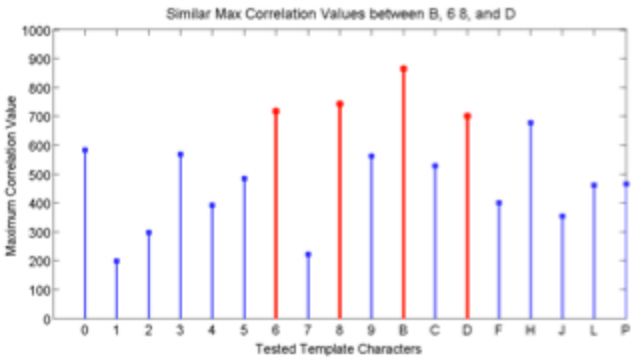
The SIFT Algorithm

The Scale Invariant Feature Transform is an important algorithm that could help improve the robustness of our program, and allow it to be expanded to new applications and environments. This algorithm can identify unique features on specific license plates, and thus be used to rescale and reorient images that are taken from a different angle. SIFT can also help the program identify different types of license plates through the different pictures and fonts between different states' license plates. Because this difference could reduce accuracy, by using SIFT a larger dictionary can be created that will be accessed when SIFT identifies a unique type of license plate.

With SIFT, our program would be able to locate and isolate a license plate from an image of an entire car and parking garage by locating target features indicative of the specific type of plate. Therefore, even if an image has many other features outside of the license plate and is captured at an abnormal angle or position, SIFT could return a license plate image that is normally aligned and identify which dictionary to access based on the type of plate.

Probabilistic Model

The matched filter program returns very similar correlation results for characters such as 'B' '6' '8' and 'D' as shown in Figure 1, which could lead to a incorrect identification. To address this issue, we can explore other means of representing or identifying each letter. One option could be running the matched filter program on different parts of characters for those with close correlation values. For example, while the whole image of 6 and 8 may look similar, the top halves of the two characters are more distinct. This could offer a second level of comparison for finer detection. Further work can also be done to create a model that utilizes probability and machine learning to choose the correct match. Through experimentation with large data sets, this method could offer a more accurate approach beyond our current deterministic model that simply chooses the highest correlation value.



Similar correlation values found for template characters 6, 8, B, and D when testing position one (character B).

Environmental Adaptation

Certain hardware and environmental adjustments can be made to maintain accuracy. Cameras should be positioned and angled for a straight, front shot of the license plate for cars of all different sizes. Lighting should be

normalized by placing a light close to the location of where the license plate will be so that the plate can still be viewed clearly at night time.

Poster
Team Poster

LICENSE PLATE RECOGNITION MATCHED FILTER TECHNIQUE

Jennifer Ding, Supreeth Mannava, Sean Peters, Omar Turk

PROJECT OBJECTIVES

Our goal is to utilize MATLAB's image processing functions to create a matched filter algorithm for License Plate Recognition and test the accuracy of our technique.

MATCHED FILTER ALGORITHM

Image Normalization
Make license plate images black and white and resize to 256x400 pixels

Partition Plate into 7 sections
Divide license plate into sections, one for each character

Rescale Pixel Values by 0.5
Shift pixel values from (0,1) to (-0.5, 0.5) to avoid multiplying by 0 in the correlation (**Figure 2**)

2D Correlation
Calculate correlation between the license plate and template dictionary of sample characters. Choose highest correlation values as correct characters

FFT FOR EFFICIENT TEMPLATE PROCESSING

$$F(p,q) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi pm/M}e^{-j2\pi qn/N} \quad \begin{matrix} p=0,1,\ldots,M-1 \\ q=0,1,\ldots,N-1 \end{matrix}$$

Our program uses MATLAB's 'conv2' function for faster image processing in the frequency domain. After flipping our image upside down and right to left, the conv2 command performs 2D correlation.

EXPERIMENTAL RESULTS



Figure 1

Original license plate image



Figure 2

Black and white image rescaled by 0.5. Black pixel values (0) becomes 0.5. White (1) becomes -0.5.

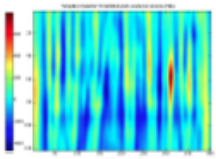


Figure 3

Correlation of license plate with template character '4' produces heat map of license plate with the correct location of '4'

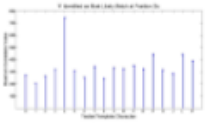


Figure 4

Correlation results of all the template characters with the sixth license plate letter

CONCLUSION AND FUTURE WORK

The application of our program to a test license plate (**Figure 1**) resulted in accurate character identification. As shown in **Figure 3** and **Figure 4**, our program can return the correct location and identity of the character.

Thus far, our program has successfully identified characters on four license plates. However, it did find very similar correlation results for 'B' and other characters (**Figure 5**) which could lead to misdetection. Further work can be done to create a probabilistic model of choosing a character rather than our current deterministic model that chooses the highest value.

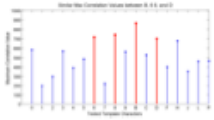


Figure 5

For future work, the addition of the Scale Invariant Feature Transform (SIFT) could provide the following features:

- Recognition of plates from different states, with different fonts through feature detection
- Realign license plate images taken at abnormal angles

ACKNOWLEDGEMENTS

We would like to thank Dr. Richard Baraniuk, Jason Holloway, and Ali Ayremlou for their contributions to the project.

License Plate Recognition Matched Filter Technique.